TROUBLESHOOTING THE PORTMASTER CONFIGURATION

This chapter describes how to analyze and evaluate issues with your PortMaster configuration. The following topics are discussed:

- How to recognize a network problem
- How to debug a network problem
- Booting from the network

RECOGNIZING NETWORK PROBLEMS

If you suspect you have a network problem there are several things you can do to try to determine the exact cause of the problem. A problem may be indicated if packets are not sent and received by the PortMaster the way you intended. Use the information in this section to troubleshoot your network.

Most of the commands described in this section can only be accessed using the command prompt interface.

VERIFYING YOUR NETWORK CONNECTIONS

You can use the Ping command to verify connectivity between your PortMaster and devices on your network. The Ping command sends an ICMP echo request to the node specified and listens for the corresponding echo reply from the specified node. If a reply is received, there is connectivity. If no reply is received there is a lack of connectivity somewhere on your network between the machine issuing the Ping request and the specified device.

If you do not receive a Ping response, check the following:

- Verify that all of the cables are connected to the PortMaster properly.
- If the machine you pinged is on another subnet, verify that you are using the correct netmask.

VERIFYING YOUR CONFIGURATION

If you have verified that everything is connected properly, you should check the configuration of your PortMaster interfaces using the ifconfig command. The ifconfig command allows you to view the active configuration of each network interface by showing the name of the interface, various flags, and other configuration information. The ifconfig flags are described in Table 12-1.

Flag	Description
IP_UP	The interface is up and running the IP protocol.
IP_DOWN	The IP protocol is not in use.
IPX_UP	The interface is up and running the IPX protocol.

IPX_DOWN	The IPX protocol is not in use.
BROADCAST	This interface is connected to an Ethernet network.
POINT_TO_POINT	The network connection on this interface is a point-to-point connection.
LISTEN	The interface is set to listen for RIP packets but not to broadcast them.
RIPSEND	RIP packets are being sent out from the interface but are not listened for.
PRIVATE	No routing information is being sent or listened to on this interface.
IFILTER	An input packet filter is set on this interface.
OFILTER	An output packet filter is set on this interface.
SUSPENDED	This interface is set for on-demand dial-out operation and is available, but does not have an active connection to the remote site.

The second and third lines of the ifconfig response contain the information described in Table 12-2.

Information	Description
inet	Indicates the Internet address of the interface.
dest	Indicates the destination Internet address of a point-to-point connection to this interface.
netmask or dest.	Indicates the netmask for the IP address shown in inet
broadcast	Indicates the broadcast address of the interface if it is an Ethernet interface.
mtu interface.	Indicates the maximum transmission unit for the
ipxnet	Indicates the IPX network number of the interface.
ipxframe	Indicates the IPX frame type for the interface.

DEBUGGING NETWORK PROBLEMS

The following subsections describe some of the things that you can do to correct network problems related to your PortMaster, once they are discovered. Most of the commands described in this section can only be accessed using the command prompt interface.

Diagnostic Mode

To force the PortMaster S0 port into diagnostic mode, follow these steps:

1. Attach a terminal to the console port s0 using a null modem cable.

For more information, refer to the Hardware Installation Guide that came with your PortMaster.

2. Raise the #1 DIP switch, left-most, on the back of the PortMaster to put the machine in Diagnostic Mode.

Refer to your Hardware Installation Guide for detailed information about the PortMaster DIP switches.

3. Cycle the power on and observe the diagnostic output.

If the PortMaster completes its diagnostics and produces a login: prompt, then the PortMaster booted correctly. If not, network booting may be required.

Note - Refer to the Hardware Installation Guide for more information on diagnostic boot messages and LED indications.

TRACING PACKETS

The ptrace command allows you to see packet information as it passes through the PortMaster. Filters are used to define which packets you want to view. The ptrace command uses the name of a filter as its argument. All packets passing through the PortMaster are evaluated against the selected filter, except UDP and ICMP packets generated by the PortMaster itself. Packets that are allowed by the filter cause a user- readable display of the following packet information:

Source of the packet

Destination of the packet

Protocol

Other protocol specific information

Filters are used to narrow the ptrace output to only those packets of interest.

If no filter is specified with the ptrace command, packet tracing is disabled.

Note - If you are using ptrace through a Telnet session, your filter should deny your Telnet packets. Otherwise, the ptrace command displays information for all of your own packets.

The following example uses a filter that denies all Telnet packets while allowing all IP traffic for evaluation

```
Command› add filter all
Command› set filter all 1 deny tcp src eq 23
Command› set filter all 2 deny tcp dst eq 23
Command› set filter all 3 permit
Command› ptrace all
```

To stop viewing packet trace information, type:

Command> ptrace

TRACING ROUTES WITH IP

You can use the traceroute command to identify all of the routers used to accesses a remote host. The traceroute command sends UDP packets to the specified host and listens for ICMP messages from routers. A host name or IP address of the destination host is entered with the traceroute command and a list of routers in the order seen is printed.

To stop the traceroute command, enter the traceroute command with no address.

RESETTING THE SYSTEM

PortMaster ports should be reset after any change to their configuration to make the new settings active. Resetting a port causes DTR to be held low for 500 milliseconds. Ports are reset when a connection drops. You can reset the whole system or specific ports using the reset command or by clicking the Reset button in PMconsole.

FORGOTTEN PASSWORDS

This section describes what to do if you have forgotten the administrative password. If you are running a ComOS version prior to 2.4 or IRX ComOS prior to 1.8R follow the instructions in "Booting from the Network" if you have forgotten your password.

If you are running ComOS version 2.4 or later or IRX ComOS version 1.8R or later, follow these steps if you have forgotten your password.

1. Place the PortMaster in diagnostic mode by raising dip switch 1.

2. Log in to the PortMaster at the PortMaster Console login: prompt using !root and a password of override.

A 16-character encrypted challenge is displayed.

3. Contact Livingston Technical Support for the appropriate 16-character encrypted response.

4. Log in to the PortMaster as !root and enter the encrypted response given by technical support as the password.

5. Change the administrative password using the set password command.

6. Type the save all command to save the new password to nonvolatile memory.

BOOTING FROM THE NETWORK

Network booting is necessary if the FLASH RAM on your PortMaster becomes corrupted. You can determine that the FLASH is corrupt if any of the following occur:

> Your PortMaster never reaches the Login: prompt during self diagnostics-when DIP switch #1 is UP.
>
> A checksum error on the ComOS is reported during the boot process.
>
> Three unsuccessful upgrade attempts on PortMasters with a ComOS of version 3.0.4 or prior or IRX ComOS version 3.0.1R or prior. In this case the ComOS has run out of file descriptors.
>
> Netbooting is also required if you have forgotten the administrative password on a PortMaster with a ComOS prior to 2.4 or IRX ComOS versions prior to 1.8R.

Note - Network booting only works if you have a host on the Ethernet that supports TFTP. Otherwise, you must boot from the PROM using the download command.

NETWORK BOOTING

If you have determined from the previous discussion that it is necessary to boot your PortMaster from the network, follow these steps:

1. Start FTP and download the appropriate generic ComOS, by typing:

```
% ftp ftp.livingston.com

Name: anonymous

Password: your email address

ftp> binary
ftp> cd pub/livingston
ftp> get README.NETBOOT
ftp> quit
```

2. Read the README.NETBOOT file to determine which net bootable operating system to download using FTP.

3. Repeat step 1 to download the appropriate GENERIC.OS.

4. If your host supports RARP on the same Ethernet segment as the PortMaster, add the Ethernet address of the PortMaster to your /etc/ethers file or your NIS map.

5. Start the rarpd service, if it is not already running, by typing:

```
% rarpd -a
```

If your system does not have RARP, use the procedures in "PROM Booting" below.

6. Set up TFTP by typing:

```
% umask 22
% mkdir /tftpboot
% mv GENERIC.OS /tftpboot/GENERIC.OS
% cd /tftpboot
% ln -s . tftpboot
```

This procedure should be done even if your host does not support RARP. If you are booting an IRX from the network, the GENERIC.OS file should be moved to /tftpboot/GENERIC.IRX.

7. Using a text editor, uncomment the tftp entry in the /etc/inetd.conf file.

8. To have the inetd daemon reread the /etc/inetd.conf file, send a SIGHUP signal to the inetd process.

9. Set the #2 DIP switch on the PortMaster to UP.

10. Boot the PortMaster and login as !root with no password.

11. If you want to save your PortMaster configuration before reformatting the FLASH RAM and your host is supported, type the following on your UNIX host:

```
% pmreadconf portmaster_name portmaster_password output_file
```

There have been occasions when something in the configuration corrupted the FLASH RAM. If this is the case, reconfigure your PortMaster from scratch after you have installed the new ComOS in FLASH RAM.

12. To erase the configuration information stored in FLASH RAM, do one of the following on the PortMaster console:

- If you are running ComOS 3.0, 3.0R, or later, type:

  ```
  Command› set register 0xffff 0x0102
  ```

  After about thirty seconds, the following message is displayed:

  ```
  Successfully formatted FLASH 2
  ```

- If you are running ComOS 2.4 or older, type:

  ```
  Command› set register 0xffff 0x0f02
  ```

  After a few moments, the following message is displayed:

  ```
  Successfully formatted FLASH 2
  ```

  Then type:

  ```
  Command› set register 0xffff 0x0f03
  ```

  After a few moments, the following message is displayed:

  ```
  Successfully formatted FLASH 3
  ```

- If you are performing this procedure because the ComOS in the FLASH RAM seems corrupted, type:

    Command› set register 0xffff 0x0f63

    After about 30 seconds, the following message is displayed:

    Successfully formatted FLASH 99

    CAUTION - This command formats all four FLASH chips, thereby removing the entire ComOS. Do not reboot the PortMaster until you reinstall the ComOS.

These procedures have reformatted the FLASH RAM on the PortMaster.

13. If you have chosen one of the noconfig files, you need to set the IP address so that you can connect to the PortMaster using the PMconsole program installed on one of your workstations, by typing:

Command› ifconfig ether0 address 192.168.200.1

In this case, 192.168.200.1 is the IP address of the PortMaster. If you are using a netmask other than 255.255.255.0 on your network, you must enter the netmask now by typing:

Command› ifconfig ether0 netmask 255.255.255.0

14. To install the new ComOS into the FLASH RAM, run PMconsole on your workstation and select the Upgrade option from the Install menu.

If you are running version 3.0.4 or later, upgrade both the ComOS and the image, in sequence.

15. Turn off the power on the PortMaster. Remove the terminal from the console port and return the DIP switches to their normal operating positions. Turn on the PortMaster power.

Everything should be working at this point. You must now reenter your configuration parameters.

PROM BOOTING

Beginning with PROM level F, a feature has been added that allows you to boot using the PROM instead of RARP. You can either boot from the tftpd daemon, or you can send a ComOS from your workstation to the console port on the PortMaster and boot from that.

If you have determined from the previous discussion that it is necessary to boot your PortMaster from PROM, follow these steps:

Note - This procedure only works with PROMs of level F or higher. The PROM version is displayed at boot time if the console port is in diagnostic mode.

1. Place the PortMaster in diagnostic mode by raising dip switch 1.

2. Attach a terminal to the console port of the PortMaster.

3. Press [ESC] or type ^[to display a > prompt.

The commands shown in Table 12-3 are now available.

PROM Command	Description
address	Allows you to set the address of the Ethernet port.
netmask	Allows you to set the netmask of the Ethernet port. Default is 255.255.255.0.
gateway	Allows you to set the default gateway in order to boot from a server on another network.
tftp	Causes the PortMaster to issue the TFTP request to the boot server.
download	Allows you to download the ComOS using the serial port.
continue	Causes the PortMaster to return to RARP mode.

4. Enter the address of the PortMaster Ethernet interface by typing:

> address 192.168.200.1

5. Set the gateway and netmask, if needed.

6. Download the appropriate generic ComOS to the workstation you want to use as the boot server, by typing:

% ftp ftp.livingston.com

Name: anonymous

Password: your email address

ftp> binary

ftp> cd pub/livingston

```
ftp> get README.NETBOOT

ftp> quit
```

7. Read the README.NETBOOT file to determine which net bootable operating system to download using FTP.

8. Repeat step 6 to download the appropriate GENERIC.OS.

9. Use one of the following to boot the PortMaster.

- To boot the ComOS from the boot server using TFTP, on the console type:

```
> tftp 192.168.200.2
```

Where 192.168.200.2 is the IP address of the TFTP host that has the GENERIC.OS software. The PortMaster then boots using the ComOS from the boot server. The new ComOS has not yet been loaded into the FLASH RAM of your PortMaster.

- To download the ComOS directly through the serial line, type:

```
> download size
```

Where size is the number of bytes of ComOS that follows. The PortMaster then boots using the ComOS downloaded from the serial connection. The new ComOS has not yet been loaded into the FLASH RAM of your PortMaster.

10. To install the new ComOS into the FLASH RAM, run PMconsole on your workstation and select the Upgrade option from the Install menu.

11. After the upgrade has completed, turn off the power on the PortMaster. Remove the terminal from the console port and return the DIP switches to their normal operating positions. Turn on the PortMaster power.

This reboots the machine. Everything should be working at this point. If not, contact Livingston Technical Support.

This example is for a hardwired network interface on port S1; if you use dial on demand you should add the filter to the appropriate location and netuser table entries (e.g. for location internet and netuser internet you would do "set internet ifilter internet.in" and "set user internet ifilter internet.in" after doing the following commands).

In this example we'll use the fictional domain example.com using the class C network 192.9.200.0, with a ftp server at ftp.example.com, a nameserver at ns.example.com, the IRX itself as gw.example.com with the service provider's router as gw.isp.net. 192.9.200.0 should be replaced by your own network number and all the hostnames should be replaced by the real hostnames or IP addresses.

```
add filter internet.in

set filter internet.in 1 deny 192.9.200.0/24 0.0.0.0/0
set filter internet.in 2 permit tcp estab
set filter internet.in 3 permit udp dst eq 53
set filter internet.in 4 permit tcp dst eq 53
set filter internet.in 5 permit tcp dst eq 25
set filter internet.in 6 permit icmp
set filter internet.in 7 permit 0.0.0.0/0 ftp.example.com/32 tcp dst eq 21
set filter internet.in 8 permit tcp src eq 20 dst gt 1023

set s1 ifilter internet.in
save all
reset s1
```

1. Block any incoming packets claiming to be from your own network
2. Allow any outgoing TCP connections
3. Allow Domain Name service queries both ways
4. Allow Domain Name service zone transfers
5. Allow mail both ways
6. Allow ICMP (ping) both ways
7. Allow anyone to FTP to our FTP host
8. Allow us to FTP things from the Internet (this is potentially risky)

If your Domain Name Server is on the outside of your local net, you need to add a line like this:
set filter internet.in 9 permit udp src eq 53
and you may then want to add an output filter like

```
add filter internet.out
set filter internet.out 1 deny 0.0.0.0/0 192.9.200.0/24
set filter internet.out 2 permit ns.example.com/32 0.0.0.0/0 tcp
set filter internet.out 3 permit ns.example.com/32 0.0.0.0/0 udp src eq 53
set filter internet.out 4 permit ns.example.com/32 0.0.0.0/0 udp dst eq 53
set filter internet.out 5 permit gw.example.com/32 gw.isp.net/32 udp dst eq 520
set filter internet.out 6 permit icmp
set s1 ofilter internet.out
save all
reset s1
```

If you want to listen for RIP information you should add:
set filter internet.in 10 permit gw.isp.net/32 gw.example.com/32 udp dst eq 520

If you want to allow auth (RFC 931) queries in (which some mailers and FTP servers use) you need to add a line like this:
set filter internet.in 10 permit tcp dst eq 113

The rules are applied in the order given, and you can either permit or deny. Anything not permitted is denied at the end.

For greater security you should further limit which hosts can do what, e.g. limit DNS and SMTP interchange with the internet to a single well-secured host of yours, and have your internal hosts refer to that host.

You can specify hosts as IP addresses or as names. You can specify subnets too; for example if we wanted to allow one subnet to have complete access to our network, we could add a rule to internet.in like:

permit 192.187.195.0/24 192.9.200.0/24

In Release 3.0 you can route and filter IPX as well, and outgoing SAP.

You can set filters on incoming packets and/or outgoing packets on each port (or ethernet). Filtering incoming packets is safer than filtering outgoing packets, because 1) you know which interface that packet is coming in on, and 2) you can protect the router itself with the filter. Other vendors' routers that only allow filtering outgoing packets are vulnerable to attack on the router itself.

EXAMPLE TWO

Here's a basic firewall filter for use with a bastion host and a IRX-111 connected to the internet on port S1.

This example is for a hardwired network interface on port S1; if you use dial on demand you should add the filter to the appropriate location and netuser table entries (e.g. for location internet and netuser internet you would do "set internet ifilter internet.in" and "set user internet ifilter internet.in" after doing the following commands).

This example allows any kind of outgoing connection from the bastion host, blocks all incoming traffic to any host but the bastion, and allows the following incoming traffic to the bastion: SMTP, NNTP, DNS, FTP, ICMP. Note that unless you have the latest versions of ftpd and sendmail you may be vulnerable to attacks through those ports.

The name bastion below should be replaced by the IP address or hostname of the bastion host.

add filter internet.in

set filter internet.in 1 deny 192.9.200.0/24 0.0.0.0/0
set filter internet.in 2 permit 0.0.0.0/0 bastion/32 tcp estab
set filter internet.in 3 permit 0.0.0.0/0 bastion/32 tcp dst eq 21
set filter internet.in 4 permit 0.0.0.0/0 bastion/32 tcp src eq 20 dst gt 1023
set filter internet.in 5 permit 0.0.0.0/0 bastion/32 tcp dst eq 119
set filter internet.in 6 permit 0.0.0.0/0 bastion/32 tcp dst eq 25
set filter internet.in 7 permit 0.0.0.0/0 bastion/32 udp dst eq 53
set filter internet.in 8 permit 0.0.0.0/0 bastion/32 tcp dst eq 53

```
set s1 ifilter internet.in
save all
reset s1

1. Block any incoming packets claiming to be from your own network
2. Allow any established TCP connections back into the bastion (you may want
       to limit this further by putting deny commands ahead of it)
3. Allow anyone to FTP to the bastion
4. Allow bastion to FTP things from the Internet (this is potentially risky)
5. Allow incoming news (NNTP) to the bastion
6. Allow incoming mail (SMTP) to the bastion
7. Allow Domain Name service queries to the bastion
8. Allow Domain Name service zone transfers from the bastion to others

The rules are applied in the order given, and you can either permit
or deny.  Anything not permitted is denied at the end.


If you have any other questions we'd be glad to answer them, send
email to support@livingston.com

--------------------------------------------------------------------------------

Subject: CERT 1/23 Advisory
Summary: What to do on IRX or PortMaster

The IRX and PortMaster discard source-routed packets but this recent
atack does not involve source routes; it spoofs the source IP address.

You can block this IP spoofing attack with your IRX (or PortMaster);
rules for doing so are included in the example in the Firewall
Application Note included with the IRX-211 or available from
ftp://ftp.livingston.com/pub/livingston/firewall/firewall-1.0.ps.Z
A short description follows.

Let's say your network is 199.9.200.0 on ether0 or ether1 or split
across both, and that your s1 sync port has an input filter called
internet.in and (optionally) an output filter called internet.out

Then insert as the VERY first rule in internet.in
deny 199.9.200.0/24 0.0.0.0/0 log

You can leave off the log if you don't want to know when you're
being attacked.  If you set a loghost, packets that match a rule
with the "log" keyword send a message to the auth.notice facility
on the loghost.

It is also useful to block packets that are trying to leave your
network but have a destination address in your network.  To do so,
insert a first rule to internet.out with
deny 0.0.0.0/0 199.9.200.0/24 log

Basically, if you *know* an address couldn't possibly be coming in via
some interface, it is useful to block it and log the event if it
happens, because it means either someone's trying to spoof you, or
something odd is happening with routing that should be looked into.

The original CERT advisory is available from
ftp://cert.org/pub/cert_advisories/CA-95:01.IP.Spoofing.Attacks.and.Hijacked.Connect
```

Subject: Re: CERT Advisory 96.21
Summary: What to do on IRX or PortMaster

Someone out on the Internet has mounted a denial-of-service attack on Panix, a relatively large ISP in the New York, USA, area. The perpetrator of this attack pretty well brought Panix to its knees for a period of almost two weeks because there was no easy way to determine from where the attack had come.

The purpose of this document is to assist Internet Service Providers and other organizations with Internet connections to configure their networks to prevent denial-of-service attacks, such as the one used against Panix, from being mounted from the ISP or organization. Preventing such attacks from occuring makes your organization a better network neighbor and may prevent one's organization from being embarassed by being the unwilling conduit for such nefarious behavior by an employee or customer.

Technical details of the attack:

The attack is to send TCP SYN packets (connection requests) to various TCP ports on servers at a rate of approximately 20 per second. The TCP on the servers can only accept so many connections at one time so legitimate users can not get a connection to the server(s). In order to hide his identity the attacker uses a random IP source address for each connection request making it very difficult to trace back the source of the attack. (It is possible to eventually trace the source of the attack but it requires tremendous effort and the cooperation of all the networks along the way between the source and destination of the attack.)

An attack of 20 connection requests per second only requires about 900 bytes per second making it possible that the attack is coming from a source with a link as slow as 9600 bps, i.e. it could be coming from a dial-up connection.

Since many of the dial-up Internet connections in the world pass through Livingston PortMasters(tm), we thought that we would assist by showing PortMaster owners how to use the features of the PortMaster to prevent spoofing-type denial-of-service attacks from originating behind a PortMaster.

The full technical details of the attack are available in greater detail in CERT(sm) Advisory CA-96.21.

A Solution:

The attack depends on the ability to "spoof" the source address thus making the attack appear as if it were coming from all over the Internet and not from where the hacker actually connects to the Internet. This effectively hides the attacker by making him appear to be everywhere at once. The "cure" is to implement anti-spoofing filters where people connect into the network so that they cannot inject packets with bogus source addresses. A hacker is less likely to mount this sort of attack if he may be readily traced as the source of the attack.

Many organizations install anti-spoofing filters at the edge of their network where it connects to the rest of the Internet. These filters prevent packets with obviously bogus source addresses, such as the organization's own network number, from reaching the rest of the organization's network. (You are filtering for bogus addresses, aren't you?) Most organizations don't think of also adding a measure of protection for the rest of the Internet by also filtering packets outbound from the organization's network to ensure that all outbound packets have valid source addresses.

Setting up filters where the users dial in:

The easiest place to provide protection is as close to the perpetrator as possible. In the case of a hacker using a dial-up connection, this means where he dials into the Internet at an ISP.

In order to make construction of filters easy, it is important to make assigned addresses fall on a bit boundary in the IP address. For instance, if an ISP is using the PortMaster PM-2e-30, the assigned addresses should fall on a 27 bit subnet boundary, thus creating a block of 30 addresses. For example, if seven PM-2e-30's are splitting the 192.168.1 subnet up to use it as assigned pools, their assigned base addresses would be 192.168.1.33, 192.168.1.65, 192.168.1.97, 192.168.1.129, 192.168.1.161, 192.168.1.193 and 192.168.1.225. The rest of this example shows the setup for the first PortMaster, the others are similar.

Now log into your PortMaster(s) and add the filter using the following set of commands. Be sure to use the proper base address for each PortMaster's set of assigned addresses.

```
add filter dial.in
set filter dial.in 1 permit 192.168.1.32/27 0.0.0.0/0
set filter dial.in 2 deny log
save all
```

Then add the following reply-item to all your RADIUS users with assigned addresses.

```
Framed-Filter-Id = "dial"
```

Here is a sample RADIUS user entry for a PPP user with an assigned address.

```
brian           Password = "UNIX"
                User-Service-Type = Framed-User,
                Framed-Protocol = PPP,
                Framed-Address = 255.255.255.254,
                Framed-Filter-Id = "dial"
```

Setting up filters at the gateway to the NSP:

Another method is to set up anti-spoofing filters outbound from the ISP where it connects to the Network Service Provider (NSP). Its simpler to implement because the filter is added only at the gateway router, but has the drawbacks that it does not prevent spoofing within the ISP itself, and does not identify the source of the spoofed packets.

Let's assume that an ISP has a T1 to its NSP and that it has been assigned a pair of class C networks for its operation. It can protect the Internet with a single filter outbound on their router. If the two networks come from a superblock (meaning the two networks may be subsumed under a single subnet mask of /23) then the outbound filter would look like this, assuming 192.168.2.0 and 192.168.3.0 and an IRX-211 with the T1 connection on S1:

```
add filter nospoof.out
set filter nospoof.out 1 permit 192.168.2.0/23 0.0.0.0/0
set filter nospoof.out 2 deny log
set s1 ofilter nospoof.out
save all
reset s1
```

If the ISP has multiple disjoint class C networks, with no supernet common to the network numbers, the filter would look like this instead:

```
add filter nospoof.out
set filter nospoof.out 1 permit 192.168.2.0/24 0.0.0.0/0
set filter nospoof.out 2 permit 192.168.7.0/24 0.0.0.0/0
set filter nospoof.out 3 permit 192.168.100.0/24 0.0.0.0/0
set filter nospoof.out 4 deny log
set s1 ofilter nospoof.out
save all
reset s1
```

Further documentation on Livingston packet filtering is available in the "Filters" chapter of the Configuration Guide for PortMaster Products and in our Firewall Application Note, ftp://ftp.livingston.com/pub/le/doc/firewall/firewall-1.1.ps